Practical Implementation of Stochastic Parameterized Model Order Reduction via Hermit Polynomial Chaos

Yi Zou Yici Cai Qiang Zhou Xianlong Hong
Sheldon X.-D. Tan and Le Kang
Department of Computer Science and Technology, Tsinghua University, Beijing
Department of Electrical Engineering, University of California, Riverside, CA

Outline

- *Introduction of Variational Model Order Reduction*
- Polynomial Chaos (PC) Representation of Input Variations
- Formulation of the Augmented System
- Efficient Implementation of the PC-SMOR
- Experimental Results

Introduction of MOR

- Model Order Reduction and Interconnect Analysis
  - MNA Equation $(G+Cs)x=Bu \quad y=L^T x$
  - Matrices G C are large, MOR reduces those large systems into smaller one, which facilitates the subsequent simulation.
  - Typical approaches
    - Projection based method
    - Balanced Truncation Reduction based method
    - Topology reduction based method
  - Static limitations

Previous Works

- MOR on Variations
  - Perturbation Parametric Approach
    - Curve Fitting and Coefficient Matching [Liu DAC99]
    - Multi-variable Taylor expansion and Two-step matching [Li ICCAD05]
  - Stochastic Approach
    - Polynomial chaos representation and reduction [Wang ICCAD04]
  - Interval arithmetic Approach
    - Change the underlying arithmetic in the MOR framework [Ma ICCAD04, ISPD05]

Wang’s work in ICCAD04

- Basics of Stochastic Parametric MOR
  - Assumes the variations of the system are stochastic
  - Treat the whole system as a stochastic process
  - Expand and represent the system via Polynomial Chaos expansion
  - Reduction could further be made on the augmented system
- Limitation
  - Lack of ability to handle Non-Gaussian variation

Support for Non-Gaussian Variation

- Reasons for supporting non-linear or Non-Gaussian variations
  - Some variations are intrinsically Non-Gaussian, (e.g leakage) or the data from actual measurement
  - Gaussian variations may generate Non-Gaussian if the operator non-linear (e.g. extraction)
- Methods to support Non-Gaussian variations
  - Multi-variable Taylor expansion
  - Polynomial Chaos expansion

Outline

- Introduction of Variational Model Order Reduction
- *Polynomial Chaos Representation of Input Variations*
- Representation of the Augmented System
- Efficient Implementation of the MOR of the Augmented System
- Experimental Results

Two representations on a log-normal variable

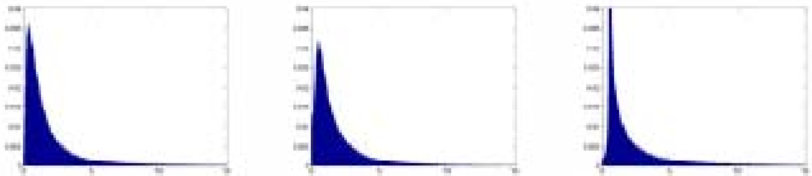
- **A log-normal variable e^{ε}**
- **Truncated Taylor expansion**

$$e^{\varepsilon} = 1 + \varepsilon + \varepsilon^2/2 + \varepsilon^3/6 + \cdots$$

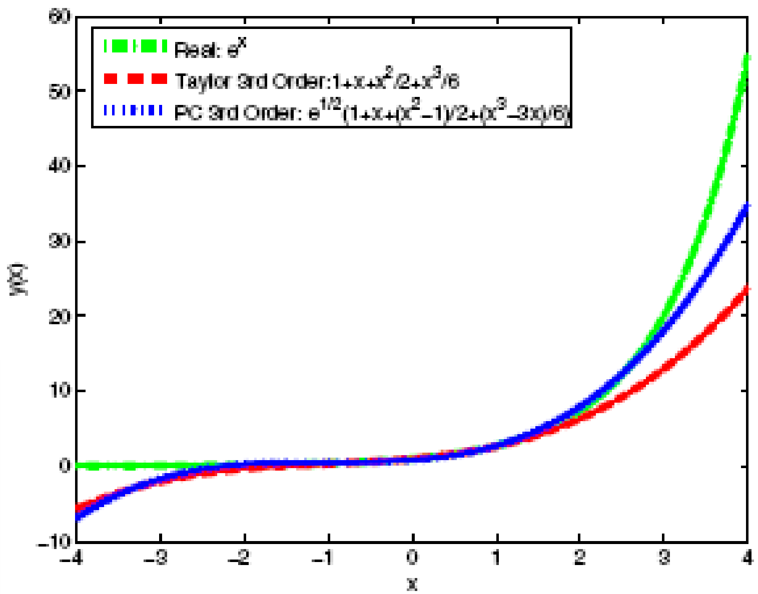
- **Polynomial Chaos expansion**
  - **Orthogonal Hermit Polynomials as bias**
  - **Determine the Coefficients under inner product**

$e^{1/2}(1+ \varepsilon +(\varepsilon^2-1)/2+(\varepsilon^3-3 \varepsilon)/6)$

(third order example)



(a) Monte Carlo (b) Taylor (3rd order) (c) PC (3rd order)



Monte Carlo		Taylor (3rd Order)		PC (3rd Order)	
mean	std	mean	std	mean	std
1.6482	2.1570	1.4999	1.7072	1.6482	2.1257

Apply PC Representation to SMOR

- Stochastic MNA equation

$$
\begin{aligned}
(\hat{G} + \hat{C}s)\hat{x} &= \hat{B}u \\
\hat{y} &= \hat{L}^T \hat{x}
\end{aligned}
$$

- Stochastic System matrix represented as affine or truncated higher order polynomial

$$
\begin{aligned}
\hat{G} &= G_0 + G_1\varepsilon_1 + G_2\varepsilon_2 + \cdots + G_n\varepsilon_n \\
\hat{C} &= C_0 + C_1\varepsilon_1 + C_2\varepsilon_2 + \cdots + C_n\varepsilon_n
\end{aligned}
$$

- Stochastic response via truncated Polynomial Chaos expansion

$$
\hat{x} = x_0 w_0 + x_1 w_1 + x_2 w_2 + \cdots + x_k w_k + \cdots
$$

- Galerkin method to minimize the truncation error

$$
\langle w_i, LHS \rangle = \langle w_i, RHS \rangle
$$

Outline

- Introduction of Variational Model Order Reduction
- Polynomial Chaos Representation of Input Variations
- *Formulation of the Augmented System*
- Efficient Implementation of the MOR of the Augmented System
- Experimental Results

Formulation of the Augmented System

- **Truncated PC representation of the system matrices**

$$\hat{G} = G_0 + G_1 w_1 + G_2 w_2 + \cdots + G_{N_w} w_{N_w}$$

$$\hat{C} = C_0 + C_1 w_1 + C_2 w_2 + \cdots + C_{N_w} w_{N_w}$$

- **Truncated PC representation of the stochastic response**

$$\hat{x} = x_0 w_0 + x_1 w_1 + x_2 w_2 + \cdots + x_k w_k + \cdots$$

- **Put those into MNA** $(\hat{G} + \hat{C}s)\hat{x} = \hat{B}u \qquad \hat{y} = \hat{L}^T \hat{x}$
  - We can get $$\sum_{i=0}^{N_w} \sum_{j=0}^{N_w} G_i x_j w_i w_j + s \sum_{i=0}^{N_w} \sum_{j=0}^{N_w} C_i x_j w_i w_j = \sum_{i=0}^{N_w} b_i w_i$$

- **Equations on the inner product**

$$\begin{aligned} & \sum_{i=0}^{N_w} \sum_{j=0}^{N_w} G_i x_j \langle w_i w_j, w_k \rangle + s \sum_{i=0}^{N_w} \sum_{j=0}^{N_w} C_i x_j \langle w_i w_j, w_k \rangle \\ = & \sum_{i=0}^{N_w} \sum_{j=0}^{N_w} G_i x_j E(w_i w_j w_k) + s \sum_{i=0}^{N_w} \sum_{j=0}^{N_w} C_i x_j E(w_i w_j w_k) \quad = \quad \sum_{i=0}^{N_w} b_i E(w_i w_k) \qquad (k = 0, 1, \cdots, N_w) \end{aligned}$$

Formulation of the Augmented System

- Block matrix form of the augmented system

$$G_{aug} = \begin{bmatrix} A_{00} & \cdots & A_{0N_w} \\ \vdots & \ddots & \vdots \\ A_{N_w0} & \cdots & A_{N_wN_w} \end{bmatrix} \qquad A_{jk} = \sum_{i=0}^{N_w} G_i E(w_i w_j w_k)$$

$$C_{aug} = \begin{bmatrix} B_{00} & \cdots & B_{0N_w} \\ \vdots & \ddots & \vdots \\ B_{N_w0} & \cdots & B_{N_wN_w} \end{bmatrix} \qquad B_{jk} = \sum_{i=0}^{N_w} C_i E(w_i w_j w_k)$$

$$b_{aug} = \begin{bmatrix} b_0 \\ \vdots \\ 0 \end{bmatrix} \quad x_{aug} = \begin{bmatrix} x_0 \\ \vdots \\ x_{N_w} \end{bmatrix} \quad L_{aug} = \begin{bmatrix} Lw_0 \\ \vdots \\ Lw_{N_w} \end{bmatrix}$$

- Properties of the augmented system
  - Block symmetric
  - Many inner products of basis functions $E(w_i w_j w_k)$ are zero
  - Sparse and diagonal dominant Matrix

Outline

- Introduction of Variational Model Order Reduction
- Polynomial Chaos Representation of Input Variations
- Representation of the Augmented System
- *Efficient Implementation of the PC-SMOR*
- Experimental Results

Efficient Implementation of the PC-SMOR

- **PRIMA like algorithm on the augmented system**
  - Other MOR reduction could also be used.
  - Note in the MOR process, in order to keep the implicit representation of the augmented system. Krylov space iteration method and Implicit Matrix-vector product are used.

$$\begin{cases} (G_{aug} + sC_{aug})* = b_{aug} \\ y = L_{aug}^T x_{aug} \end{cases}$$

$$G_{aug} = \begin{bmatrix} A_{00} & \cdots & A_{0N_w} \\ \vdots & \ddots & \vdots \\ A_{N_w0} & \cdots & A_{N_wN_w} \end{bmatrix}$$

$$C_{aug} = \begin{bmatrix} B_{00} & \cdots & B_{0N_w} \\ \vdots & \ddots & \vdots \\ B_{N_w0} & \cdots & B_{N_wN_w} \end{bmatrix}$$

$$b_{aug} = \begin{bmatrix} b_0 \\ \vdots \\ 0 \end{bmatrix} \quad x_{aug} = \begin{bmatrix} x_0 \\ \vdots \\ x_{N_w} \end{bmatrix} \quad L_{aug} = \begin{bmatrix} Lw_0 \\ \vdots \\ Lw_{N_w} \end{bmatrix}$$

1.start from the augmented system $G_{aug}, C_{aug}, b_{aug}$ in (16).
2.Solve $G_{aug}X_0 = b$.
3.Orthogonalize and normalize each column in X_0.
For $k = 1, 2, \cdots$
    4.Solve $G_{aug}X_k = C_{aug}X_{k-1}$
    5.Orthogonalize and normalize each column in X_k
end for
6.$X = [X_0 X_1 \cdots X_k]$.
7.Obtain the reduced system matrix from
$\tilde{G} = X^T G_{aug} X, \tilde{C} = X^T C_{aug} X,$
$\tilde{b} = X^T b_{aug}, \tilde{L} = X^T L_{aug}$.
8.Perform the eigen decomposition $\tilde{G}^{-1}\tilde{C} = S\Lambda S^{-1}$
9.Obtain the poles, residues and transfer functions

Efficient Implementation of the PC-SMOR

- **Implicit Matrix-vector product**

$$\begin{bmatrix} A_{00} & \cdots & A_{0N_w} \\ \vdots & \ddots & \vdots \\ A_{N_w0} & \cdots & A_{N_wN_w} \end{bmatrix} * \begin{bmatrix} x_0 \\ \vdots \\ x_{N_w} \end{bmatrix} = \begin{bmatrix} r_0 \\ \vdots \\ r_{N_w} \end{bmatrix}$$

$$r_i = \sum_{j=0}^{N_w} A_{ij} * x_j = \sum_{i=0}^{N_w} \sum_{j=0}^{N_w} G_i * x_j * E(w_i w_j w_k)$$

- **Iterative methods by cases**
  - CG for symmetric case (e.g RC)
    - Might not work for very large variation, not S.P.D
  - GMRES for non-symmetric case
- **Precondition by cases**
  - Two-level Preconditioner
    - Block Jacobi at higher (Block) level
    - ILUTP at lower (matrix) level
  - Efficient to construct the preconditioner, which only needs one ILUTP on the diagonal blocks or even one block of diagonals

Outline

- Introduction of Variational Model Order Reduction
- Polynomial Chaos Representation of Input Variations
- Representation of the Augmented System
- Efficient Implementation of the MOR of the Augmented System
- *Experimental Results*

Experimental Results

- **Tool implemented using C++**
  - Sparse Matrix Library used : GMM++
  - Other Library used: SuperLU, Lapack
- **Testing Environment**
  - AMD2000+, 768M Mem, GNU/Linux
- **Test Circuit**
  - RC and RLC, node size from 113 to 5452
  - Larger size is OK for the tool , but we want MC and direct solving still could give results
- **Variation tested:**
  - Log-normal variation, four variables tested
  - Gaussian one is similar to [Wang ICCAD04] thus we omit those
- **Expansion order of PC used**: 2
- **Sampling number of Monte Carlo**: 4000

Experimental Results

RUNNING-TIME AND MEMORY CONSUMPTION COMPARISON OF DIFFERENT METHODS

Monte Carlo		Morstat(Direct)		Morstat(Iterative)	
time(s)	mem(M)	time(s)	mem(M)	time(s)	mem(M)
78m35s	0.7	2m11s	8.7	1m37s	2.2
325m44s	1.8	14m11s	84.7	5m15s	13.2
956m9s	6.4	31m24s	231.8	11m47s	45.1

ACCURACY COMPARISON OF DIFFERENT METHODS

Monte Carlo		Morstat(Taylor)		Morstat(PC)	
mean(ns)	std(ns)	mean(ns)	std(ns)	mean(ns)	std(ns)
12.22	1.719	12.13	1.724	12.23	1.721
19.75	1.921	19.55	1.911	19.77	1.923
25.86	2.412	25.82	2.414	25.88	2.409

Experimental Results

- Generally 50X faster than Monte Carlo.
- Iterative method which leverage the implicit representation uses less memory (1/Nw) and runs 2~5 faster than direct solving using SuperLU.
- PC representation on input variation also give more accurate delay results than that use Taylor expansion as in [Wang ICCAD04]

Conclusion

- Represent Non-Gaussian Input variation using PC rather than Taylor gives more accurate results
- Gives out the derivation of the augmented system and give out an Implicit form of the augmented system
- Implementation with implicit form saves much memory also very efficient comparing with original method

That’s all. Thanks for Listening.

- Q&A